Importance of Search and Evaluation Strategies in Neural Dialogue Modeling

Ilia Kulikov
New York University
kulikov@cs.nyu.edu

Alexander H. Miller
Facebook AI Research

Kyunghyun Cho
New York University
Facebook AI Research
CIFAR Azrieli Global Scholar

Jason Weston
Facebook AI Research
New York University

Abstract

We investigate the impact of search strategies in neural dialogue modeling. We first compare two standard search algorithms, greedy and beam search, as well as our newly proposed iterative beam search which produces a more diverse set of candidate responses. We evaluate these strategies in realistic full conversations with humans and propose a model-based Bayesian calibration to address annotator bias. These conversations are analyzed using two automatic metrics: log-probabilities assigned by the model and utterance diversity. Our experiments reveal that better search algorithms lead to higher rated conversations. However, finding the optimal selection mechanism to choose from a more diverse set of candidates is still an open question.

1 Introduction

There are three high-level steps to building a neural autoregressive sequence model for dialog modeling, inspired by work of Vinyals and Le (2015). First, decide on a network architecture which will consume previous utterances as well as any extra information such as speaker identifiers. Second, choose a learning strategy. Finally, decide on a search algorithm, as neural autoregressive sequence models do not admit a tractable, exact approach for generating the most likely response.

Recent research in neural dialogue modeling has often focused on the first two aspects. A number of variants of sequence-to-sequence models (Sutskever et al., 2014; Cho et al., 2014; Kalchbrenner and Blunsom, 2013) have been proposed for dialogue modeling in recent years, including hierarchical models (Serban et al., 2016) and transformers (Mazaré et al., 2018; Yang et al., 2018). These advances in network architectures have often been accompanied by advanced learning algorithms. Serban et al. (2017) introduce latent variables to their earlier hierarchical model and train it to maximize the variational lower bound, similar to Zhao et al. (2017) who propose to build a neural dialogue model as a conditional variational autoencoder. Xu et al. (2017) and Li et al. (2017b) train a neural dialogue model as conditional generative adversarial networks (Mirza and Osindero, 2014). These two learning algorithms, variational lower-bound maximization and adversarial learning, have been combined into a single model by Shen et al. (2018), which has been followed by Gu et al. (2018).

Despite abundant endeavors on modeling and learning, search has received only a little attention (Dinan et al., 2019). Most of the work on search has focused on training an additional neural network that provides a supplementary score to guide either greedy or beam search. Li et al. (2015) propose a maximum mutual information criterion for decoding using a reverse model. This has been extended by Li et al. (2017a), where an extra neural network is trained to predict an arbitrary reward given a partial hypothesis and used during decoding. Similarly, Zemlyanskiy and Sha (2018) train a neural network that predicts the other participant's personality given a partial conversation and use its predictability as an auxiliary score for re-ranking a set of candidate responses. None of these approaches study how the choice of the underlying search algorithm, rather than its scoring function, affects the quality of the neural dialogue model.

In this paper, we investigate the effects of varying search and selection strategies on the quality of generated dialogue utterances. We start with an attention-based sequence-to-sequence model (Bahdanau et al., 2014) trained on the recently-released PersonaChat dataset (Zhang et al., 2018). We evaluate three search algorithms: greedy search, beam search and iterative beam search, the last of which we design based on earlier works by Batra et al. (2012). These algorithms are qualitatively different from each other in the size of subspace over which they search for the best response.

We compare all of these alternatives using two families of metrics. First, we use human evaluation of full, multi-turn conversation. The resulting distribution of annotator's scores has huge variance that is rarely discussed nor analyzed by other groups. This variance comes from each annotator's individual attitude towards and understanding of the task, which we call annotator bias. In order to address this bias, we propose model-based Bayesian calibration that explicitly factors in each annotator's bias and the algorithm's underlying score, and report the posterior mean and variance of each algorithm's score. Additionally, we also compare automatic metrics that capture the model's intrinsic preference (log-probability) and the diversity of responses (distinct-n).

We make two key observations from the experiments. A better search strategy can indeed generate responses that are both intrinsically preferred by the underlying model and diverse, without re-designing or re-training the neural dialogue model. However, this observation does not necessarily carry over to human evaluation, as the best performing strategy according to these automatic metrics was not the best strategy according to human annotators. These results highlight both the importance of search algorithms as well as the difficulty in evaluating neural dialogue systems in a realistic, full conversation setup.

We will make trained models, code and human evaluation transcripts publicly available. Randomly sampled transcripts for each strategy are available in 2 additional pages of examples. All transcripts are given in additional materials and we encourage everyone to read it.

2 Neural dialogue modeling

Since Vinyals and Le (2015), a neural autoregressive sequence model based on sequence-to-sequence models Sutskever et al. (2014); Cho et al. (2014) have become one of the most widely studied approaches to dialogue modeling (see, e.g., Serban et al., 2016, 2017; Zhao et al., 2017; Xu et al., 2017; Li et al., 2016, 2017a,b; Zemlyanskiy and Sha, 2018; Zhang et al., 2018; Miller et al., 2017; Shen et al., 2018; Gu et al., 2018). In this approach, a neural sequence model is used to model a conditional distribution over responses given a context which consists of previous utterances by both itself and a partner in the conversation as well as any other information about the speaker.

2.1 Neural autoregressive sequence modeling

A neural autoregressive sequence model learns the conditional distribution over all possible responses given the context. Each conditional distribution is modelled by a neural network, and popular choices include recurrent neural networks (Mikolov et al., 2010; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014), convolutional networks (Dauphin et al., 2016; Gehring et al., 2017) and self-attention (Sukhbaatar et al., 2015; Vaswani et al., 2017). We explore search strategies and fix the model to a recurrent neural network.

Learning: Maximum Likelihood Each example in a training set D consists of auxiliary information or context U (such as a persona profile or external knowledge context) and a sequence of utterances, each of which is marked with a speaker tag, i.e., $C = (U, (Y_1^a, Y_1^b, \ldots, Y_L^a, Y_L^b) \in D$, where Y_l^s is the utterance from the l-th turn by a speaker s. The conditional log-probability assigned to this example given by a neural sequence model is then written as

$$\log p(C) = \sum_{s \in \{a,b\}} \sum_{l=1}^{L} \log p(Y_l^s | Y_{<l}^s, Y_{\leq l}^{\bar{s}}, U), \tag{1}$$

where $\bar{s} = a$ if $s = b$ and otherwise $\bar{s} = b$.

Learning maximizes the log-probabilities of all the conversations in the training set:

$$L = \frac{1}{|D|} \sum_{C \in D} \log p(C), \tag{2}$$

often done using stochastic gradient descent with backpropagation (Rumelhart et al., 1985).

2.2 Inference (generation)

In this paper, we generate a response to the current state of the conversation (but do not attempt to plan ahead to future exchanges), maximizing

$$p(Y|Y_{<l}^s, Y_{<l}^{\bar{s}}, U) = \prod_{t=1}^{T} \log p(y_t | y_{<t}, Y_{<l}^s, Y_{<l}^{\bar{s}}, U).$$

Unfortunately, it is intractable to solve this problem due to the exponentially-growing space of all possible responses w.r.t. the maximum length T. It is thus necessary to resort to approximate search algorithms.

Greedy search Greedy search has been the search algorithm of choice among the recent papers on neural dialogue modeling (Gu et al., 2018; Zhao et al., 2017; Xu et al., 2017; Weston et al., 2018; Zhang et al., 2018). It moves from left to right selecting one token at a time, simply choosing the most likely token at the current time step:

$$\hat{y}_t = \arg\max_{v \in V} \log p(y_t = v | \hat{y}_{<t}, Y^s_{<l}, Y^{\bar{s}}_{<l}, U).$$

Greedy search has been found significantly suboptimal within the field of machine translation (see, e.g., Table 1 in Chen et al., 2018), where similar neural sequence models are frequently used.

Beam search Instead of maintaining a single hypothesis at a time, as in greedy search above, at time step t beam search maintains a set of K hypotheses $\mathcal{H}_t$:

$$\mathcal{H}_t = \{(y^1_1, \ldots, y^1_t), \ldots, (y^K_1, \ldots, y^K_t)\}. \quad (3)$$

Each hypothesis $h^i_{y^i_t}, i \in \{1, \ldots, K\}$ from $\mathcal{H}_t$ is expanded with all possible next tokens v from the vocabulary V to form candidate hypotheses. Each candidate is in the form of

$$\tilde{h}^i_v = h^i_{y_t} \| (v) = (y^i_1, \ldots, y^i_t, v), \quad (4)$$

and is assigned a score:

$$s(\tilde{h}^i_v) = s(h^i_{y^i_t}) + \log p(v | y^i_{\leq t}). \quad (5)$$

The new hypothesis set of K hypotheses is then constructed as

$$\mathcal{H}_{t+1} = \underset{i,v}{\text{arg-top-}k} \;\; s(\tilde{h}^i_v). \quad (6)$$

From the new hypothesis set, we find and copy *finalized* hypotheses (sequences ending with the special token $\langle\text{eos}\rangle$ for "end of sequence") to a candidate sequence set $\mathcal{M}_t$. That is,

$$\mathcal{M}_t = \left\{ h^i_v \in \mathcal{H}_{t+1} | v = \langle \text{eos} \rangle \right\}.$$

Beam search terminates when $| \cup^t_{t'=1} \mathcal{M}_t| \geq K'$, where K' is the maximum number of candidate sequences to be returned, or when $t \geq L_{\max}$, where $L_{\max}$ is the maximum length allowed for each candidate sequence. When terminated, beam search returns all the candidate sequences in $\mathcal{M} = \cup^t_{t'=1} \mathcal{M}_t$.

One can increase the size of the subspace over which beam search searches for a response and size of $\mathcal{M}$ by changing hyper-parameters $K, K', L_{\max}$. However, beam search is known to suffer from the problem that most of the hypotheses discovered in $\mathcal{M}$ are near each other in the response space (Li et al., 2016, 2015). For tasks such as dialogue modeling, which are much more open-ended than e.g. machine translation, this is particularly troublesome as many high quality responses may be missing in the beam.

Final sequence selection We consider search strategies to produce a set of candidate responses for the model to choose from. While greedy search provides only a single possible sequence, beam search generates a candidate set of size $|\mathcal{M}|$. It is usual practice to use the score $s(h)$ used during the search to select the final sequence, but it is an open question whether there are better selection strategies for choosing between these final candidate responses.

Avoiding repeating n-grams Although this has not been reported in a formal publication in the context of neural dialogue modeling to our knowledge, Paulus et al. (2017) and Klein et al. (2017) implement so-called n-gram blocking. In n-gram blocking, a hypothesis in a beam $\mathcal{H}_t$ is discarded if there is an n-gram that appears more than once within it.

3 Uncovering hidden responses

We now propose an improved search strategy. To address the locality issue in beam search, we propose an iterative beam search to radically increase the size of search space without introducing much computational overhead, inspired by earlier work on diverse beam search (Vijayakumar et al., 2018; Batra et al., 2012; Li et al., 2016).

3.1 Iterative beam search

The search space over which beam search has operated can be characterized by the union of all partial hypothesis sets $\mathcal{H}_t$ in Eq. (3): $\mathcal{S}_0 = \cup^T_{t=1} \mathcal{H}_t$, where we use the subscript 0 to indicate that beam search has been done without any other constraint. Re-running beam search with an increased beam width K would result in the search space that overlaps significantly with $\mathcal{S}_0$, and would not give us much of a benefit with respect to the increase in computation.

Instead, we keep the beam size K constant but run multiple iterations of beam search while ensuring that any previously explored space $\bar{\mathcal{S}}_{<l} = \cup_{l'=0}^{l-1} \mathcal{S}_{l'}$ is *not* included in a subsequent iteration of beam search. This is done by setting the score of each candidate hypothesis $s(\tilde{h}_{t+1}^{i})$ in Eq. (5) to negative infinity, when this candidate is included in $\bar{\mathcal{S}}_{<l}$. We relax this inclusion criterion by using a non-binary dissimilarity metric, and say that the candidate is included in $\bar{\mathcal{S}}_{<l}$, if

$$\min_{h \in \bar{\mathcal{S}}_{<l}} \Delta(\tilde{h}_{t+1}^{i}, h) < \epsilon, \qquad (7)$$

where Δ is a string dissimilarity measure, such as Hamming distance used in this work, and ϵ is a similarity threshold.

This procedure ensures that a new partial hypothesis set of beam search in the l-th iteration minimally overlaps with any part of the search space explored earlier during the first $l-1$ iterations of beam search. By running this iteration multiple times, we end up with a set of top hypotheses from each iteration of beam search, from which the best one is selected according to for instance the log-probability assigned by the model. We build a final candidate set $\mathcal{M}$ as a set of all these best hypotheses from beam search iterations.

Practical implementation A major issue with iterative beam search in its naive form is that it requires running beam search multiple times, when even a single run of beam search can be prohibitively slow in an interactive environment, such as in dialogue generation. We address this computational issue by performing these many iterations of beam search in parallel simultaneously. At each time step in the search, we create sets of candidate hypotheses for all iterations in parallel, and go through these candidate sets in sequence from the $(l=0)$-th iteration down to the last iteration, while eliminating those candidates that satisfy the criterion in Eq. (7). We justify this parallelized approach by defining the similarity measure Δ to be always larger than the threshold ϵ when the previous hypothesis h is longer than $\tilde{h}_{t+1}^{i}$ in Eq. (7).

4 Dialogue evaluation

Broadly there are two ways to evaluate a neural dialogue model. The first approach is to use a set of (often human generated) reference responses and compare a single generated response against them (Serban et al., 2015; Liu et al., 2016). There are several methods for this comparison: (1) measure the perplexity of reference responses using the neural dialogue model, (2) compute a string match-based metric of a generated response against reference responses, and (3) use human annotators to compare model generated responses against reference or other models' responses. None of these approaches capture the effectiveness of a neural sequence model in conducting a full conversation, because the model responses are computed given a human-written context, i.e., it does not see its own responses in the dialogue history, but gold responses only.

We concentrate on a second approach for evaluation where a neural dialogue model has a multi-turn conversation with a human partner (or annotator) (Zhang et al., 2018; Zemlyanskiy and Sha, 2018; Weston et al., 2018; Dinan et al., 2019). Unlike other approaches, it requires active human interaction, as a conversation almost always deviates from a previously collected data even with the same auxiliary information (U in Eq. (1)). This evaluation strategy reflects both how well a neural dialogue model generates a response given a correct context as well as how well it adapts to a dynamic conversation—the latter was not measured by the first strategy, where the model only had to generate a single response.

4.1 Human evaluation of a full conversation

An annotator is asked to make a conversation with a randomly selected model (search strategy) for at least five turns. At the end of the conversation, we ask the annotator three sets of questions:

1. Overall score ($\{1, 2, 3, 4\}$)
2. Marking of each *good* utterance-pair ($\{0, 1\}$)
3. Marking of each *bad* utterance-pair ($\{0, 1\}$)

The first overall score allows us to draw a conclusion on which algorithm makes a better conversation overall. We use a 4 point scale in order to avoid having a "catch-all" category in the answer (Dalal et al., 2014). The latter two questions are collected to investigate the relationship between the overall impression and the quality of each utterance-pair.

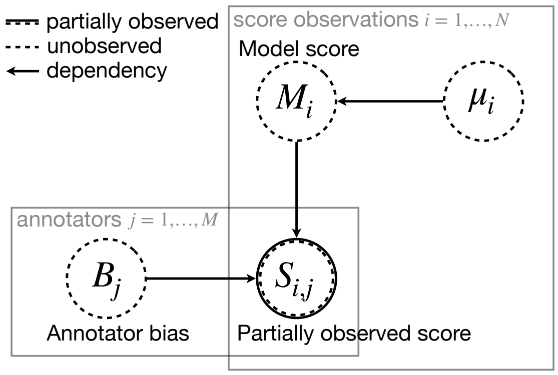

Figure 1: Graphical model used for Bayesian Calibration. M annotators participated such that in total N observed scores are presented.

4.2 Bayesian calibration

Although human evaluation is desirable, raw scores collected by annotators are difficult to use directly due to the annotator bias. Some are more generous while others are quite harsh, as recently reported in Zhang et al. (2018); Zemlyanskiy and Sha (2018). We propose using Bayesian inference as a framework to account for the bias of each annotator, and describe two instances of this framework below.

1-4 star rating of a conversation We treat both the unobserved score M_i of each model, in our case each search algorithm, and the unobserved bias B_j of each annotator as latent variables. The score of the i-th model follows the following distribution: $\mu_i \sim \mathcal{U}(1, 4)$ and $M_i \sim \mathcal{N}(\mu_i, 1^2)$, where $\mathcal{U}$ and $\mathcal{N}$ are uniform and normal distributions. It states that *a priori* each model is likely to be uniformly good or bad. The annotator bias B_j follows $B_j \sim \mathcal{N}(0, 1^2)$, where we are assuming that each annotator does not have any bias *a priori*.

Given the model score M_i and annotator bias B_j, the conditional distribution over an observed score S_{ij} given by the j-th annotator to the i-th model is then:

$$S_{ij} \sim \mathcal{N}(M_i + B_j, 1^2).$$

Due to the nature of human evaluation, only a few of S_{ij}'s are observed. Figure 1 shows the graphical model described above.

The goal of inference in this case is to infer the posterior mean and the variance:

$$\begin{aligned} &\mathbb{E}[M_i | \{S_{ij} | S_{ij} \in \mathcal{O}\}], \\ &\mathbb{V}[M_i | \{S_{ij} | S_{ij} \in \mathcal{O}\}], \end{aligned} \tag{8}$$

where $\mathcal{O}$ is a set of observed scores.

Binary rating of an utterance When an annotator labels pairs of utterances from the conversation with a binary score $\{0, 1\}$ (such as whether that pair was a "good" exchange), we need to further take into account the turn bias T_k: $T_k \sim \mathcal{N}(0, 1^2)$. As we will use a Bernoulli distribution for each observed score rather than a 1-4 rating, we modify the prior of the model scores accordingly: $M_i \sim \mathcal{N}(0, 1^2)$.

The distribution of an observed utterance-pair score is then $S_{ijk} \sim \mathcal{B}(\text{sigmoid}(M_i + B_j + T_k))$, where $\mathcal{B}$ is a Bernoulli distribution. The goal of inference is then to compute

$$\begin{aligned} &\mathbb{E}_{M_i | \{S_{ijk} | S_{ijk} \in \mathcal{O}\}} [\text{sigmoid}(M_i)] \,, \\ &\mathbb{V}_{M_i | \{S_{ijk} | S_{ijk} \in \mathcal{O}\}} [\text{sigmoid}(M_i)] \,, \end{aligned} \tag{9}$$

which estimate the average number of positively labelled utterance-pairs given the i-th model and the uncertainty in this estimate, respectively.

Inference We use no-u-turn (NUTS) sampler (Hoffman and Gelman, 2014) for posterior inference in Pyro (Bingham et al., 2018).

5 Experiment Settings

5.1 Data: Persona-Chat

We use Persona-Chat, released recently by Zhang et al. (2018) and the main dataset for the Conversational Intelligence Challenge 2 (ConvAI2),[1] to train a neural dialogue model. The dataset contains dialogues between pairs of speakers randomly assigned personas from a set of 1,155, each consisting of 4-5 lines of description about the part they should play, e.g. *"I have two dogs"* or *"I like taking trips to Mexico"*. The training set consists of 9,907 such dialogues where pairs of partners play their roles, and a validation set of 1,000 dialogues. The ConvAI2 test set has not been released. Each dialogue is tokenized into words, resulting in a vocabulary of 19,262 unique tokens. See Zhang et al. (2018) for more details.

[1] http://convai.io/

5.2 Neural dialogue modeling

Model We closely follow Bahdanau et al. (2014) in building an attention-based neural autoregressive sequence model. The encoder has two bidirectional layers of 512 LSTM (Hochreiter and Schmidhuber, 1997) units each direction-layer, and the decoder has two layers of 512 LSTM units each. We use global general attention as described by Luong et al. (2015). We use the same word embedding matrix for both the encoder and decoder, which is initialized from 300-dimensional pretrained GloVe vectors (Pennington et al., 2014) for the 97% of the vocabulary which overlaps with GloVe. We allow word embedding weights to be updated during the training.

Learning We use Adam (Kingma and Ba, 2014) with the initial learning rate set to 0.001. We apply dropout (Srivastava et al., 2014) between the LSTM layers with the dropout rate of 0.5 to prevent overfitting. We train the neural dialogue model until it early-stops on the validation set.[2]

The perplexity of trained model on the ConvAI2 validation set is 24.84, which is competitive compared to the other entries on the competition's leaderboard.[3] Our model serves well as an underlying system for investigating the effect of search algorithms.

5.3 Search Strategies

We test three search strategies; **greedy** and **beam** search from §2.2, and iterative beam search (**iter-beam**) from §3.1.

Beam search (**beam**) uses beam size $K = 5$ and $K' = 15$. This decision is based on preliminary experiments where we found that smaller beam sizes work better than larger ones do. We use the length penalty, described by Wu et al. (2016) and n-gram blocking from §2.2.

Iterative beam search (**iter-beam**) uses 15 iterations of beam search with beam size 5 resulting in a candidate set of size 15. We use the same length penalty and n-gram blocking as in beam search (**beam**). Given the hyper-parameters above both **beam** and **iter-beam** produce 15 candidates and selects the final response based on log-probability.

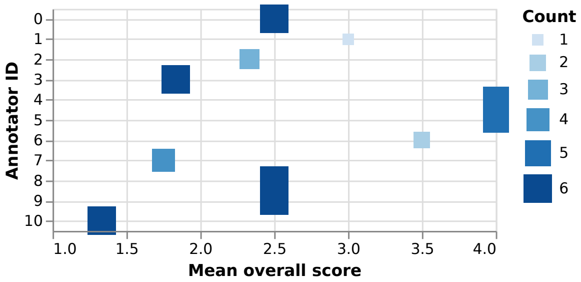

Figure 2: The distribution of averaged overall scores given by annotators to greedy search (**greedy**). Each row plots scores given by a single annotator over multiple conversations. Counts show how many dialogues each annotator performed.

5.4 Evaluation

Human evaluation We use ParlAI (Miller et al., 2017) which provides seamless integration with Amazon Mechanical Turk (MTurk) for human evaluation. A human annotator is paired with a model with a specific search strategy, and both are randomly assigned personas out of a set of 1,155, and are asked to make a conversation of at least either five or six turns (randomly decided). We allow each annotator to participate in at most six conversations per search strategy and collect approximately 50 conversations per search strategy and additional human-human test.[4] Each conversation is given a single overall score and two sequences of binary utterance pairs flags, as described in §4.1.

Bayesian calibration In order to remove annotator bias, or inter-annotator variability, we use Bayesian calibration from §4.2. We take 50 warm-up steps and collect 150 samples using NUTS sampler for inferring the posterior mean and variance of the model score in Eq. (8). We use 30 warm-up steps and 100 samples for inferring the mean and variance of the average portion of positively or negatively labelled utterance-pairs in Eq. (9).[5]

Automatic metrics In addition to human evaluation, we compute automatic metrics to quantitatively characterize each search algorithm. First, we report the **log-p**robability of a generated response assigned by the model which is a direct

[2] When the validation loss does not improve for twelve epochs, we early-stop.

[3] `https://github.com/DeepPavlov/convai/blob/master/leaderboards.md`

[4] Some conversations were dropped due to technical errors, resulting in total 50, 51, 49 and 53 conversations for **greedy**, **beam**, **iter-beam** and **humans**, respectively.

[5] Variances of inferred posterior distribution and original data distribution are not comparable, as the former reflects the uncertainty in posterior inference rather than the spread of scores.

Search strategy	log-p↑	Overall Score (1-4)↑		% Good Pairs↑		% Bad Pairs↓	
		Raw	Calibrated	Raw	Calibrated	Raw	Calibrated
greedy	-9.66±2.73	2.56±0.98	2.30±0.24	0.45	0.28±0.07	0.38	0.54±0.07
beam	-7.26±2.28	2.67±0.86	2.70±0.27	0.58	0.44±0.08	0.35	0.27±0.01
iter-beam	**-5.95±1.35**	2.80±0.90	2.67±0.23	0.58	0.45±0.08	0.31	0.32±0.03
human	-42.95±18.87	3.62±0.71	3.37±0.22	0.76	0.76±0.06	0.07	0.04±0.003

Table 1: The average log-probabilities and model scores (average±standard deviation) assigned to the responses during human evaluation. Better search algorithms find responses with higher log-probabilities according to the model. Without observing standard deviations and calibrated scores one can make erroneous conclusions.

indicator of the quality of a search algorithm. Second, we compute the average number of unique n-grams generated per conversation normalized by the number of generated tokens in the conversation, called **distinct-n** from (Li et al., 2015), with $n = 1, 2, 3$.

We compute distinct-n in two different settings. First, we compute distinct-n over the candidate set $\mathcal{M}$ given by the search algorithm. Second, we compute distinct-n over the final selected responses for each search strategy. The former shows diversity within the possible response candidates, while the latter shows diversity among the actual selected dialogue outputs.

6 Result

6.1 Human Evaluation

Annotator bias In Fig. 2, we plot the averaged scores provided by the human annotators for one search strategy (**greedy**), where each row corresponds to each annotator. Consider the three annotators with id $3, 4, 10$. Their means are clearly separated from each other, which points to the existence of annotator bias. This observation supports the necessity of the Bayesian calibration described in §4.2.

Human evaluation In Table 1, we present the scores from human evaluation. In total 41 unique annotator participated within 201 collected conversations. We make a major observation which is that greedy search (**greedy**), which has been the search algorithm of choice in neural dialogue modeling, significantly lags behind the variants of beam search (**beam, iter-beam**) in all metrics. This stark difference is worth our attention, as this difference is *solely* due to the choice of a search algorithm and is not the result of different network architectures nor learning algorithms. In fact, this cannot even be attributed to different parameter initialization, as we use only *one* trained model for all of these results.

The model scores assigned to human conversations (**humans**) are far superior to all search strategies. It is clear with both overall score and utterance pairs proportion scores. This tells us that there are many open questions how to improve neural dialogue models.

6.2 Automatic Metrics

Search quality: log probability (log-p) Better search algorithms find responses with higher log-probability according to the model, as shown in Table 1. This is a natural consequence from exploring a larger subset of the search space.

A notable observation from Table 1 is that the neural sequence model assigns very low log-probabilities to human response. This implies that there is a limit to improving this specific neural dialogue model by using a better search algorithm and instead there is more room to improve the models and learning algorithms to place a high probability on human responses. It is necessary to test the proposed search strategies with new models and we leave this for the future.

Diversity: distinct-n The diversity metric is measured before (pre) and after (post) selecting the final response from the candidate set $\mathcal{M}$ for both **beam** search and **iter-beam** search. Since **greedy** and **humans** produce only single response, we compute the diversity metric only using those final responses for both greedy search and humans. In both pre and post settings, the normalization is done per each conversation.

As well as in human evaluation, **greedy** has lower diversity compared to all the other strategies as shown in Table 2. We see large gap in pre-selection distinct-n for all n between **beam** and **iter-beam** while the difference is small in post-selection distinct-n. In other words, while providing more diverse set of candidates, the final selected output response with **iter-beam** is not

	distinct-n ↑					
Search	$n = 1$		$n = 2$		$n = 3$	
strategy	post	pre	post	pre	post	pre
greedy	0.47	-	0.61	-	0.62	-
beam	0.56	0.06	**0.69**	0.12	**0.63**	0.18
iter-beam	**0.59**	**0.18**	0.68	**0.41**	0.60	**0.58**
human	0.66	-	0.85	-	0.82	-

Table 2: Measuring the diversity of different search and selection strategies. Both **beam** and **iter-beam** produce up to 15 hypotheses each. Column **post** is distinct-n measured over the final selected output responses given by the model and allows us to compare the diversity of the best responses each search procedure produces. Column **pre** is distinct-n measured over the candidate set $\mathcal{M}$ given by the search algorithm and allows us compare diversity generated within the search.

Beam search	**Iterative beam search**
do you have any pets ?	that ’ s cool , what do you like to eat ?
what is your favorite animal ?	do you have a favorite color ? mine is pink .
i like to talk about strangers .	what do you like to eat ?
do you like animals ?	i don ’ t like fish , but my favorite color is pink .
do you like animals ? i want to live at the beach .	what color is your hair ?
do you like animals ? i ’ ve a pet .	that does sound good , i like to go alone .
do you like animals ? i want to live on the beach .	i would love to eat fish .
do you like animals ? i want to live at the beach	that makes sense , do you have any hobbies ?
do you like animals ? i ’ ve a monkey .	i hear you , my favorite color is pink .
do you like animals ? i want to be a monkey .	i want to be a yoga instructor .
do you like animals ? i want to live at the beach , but love it .	i did not eat meat , but my favorite color is pink .
do you like animals ? i want to live at the beach , but love monkeys .	what are your favorite foods ? mine is pink .
do you like animals ? i want to live at the beach , but they are my favorite .	what does your favorite color ? mine is pink .
do you like animals ? i want to live at the beach , but have a monkey .	what type of food do you like ?
do you like animals ? i want to live at the beach , but they are my favorite	i spend a lot of time alone .

Table 3: **beam** and **iter-beam** candidate sets $\mathcal{M}$. These are from one turn of one randomly selected conversation from human evaluation. **iter-beam** produces more diverse responses.

particularly diverse. This agrees well with human evaluation, where both **iter-beam** and **beam** model scores were indistinguishable, as annotators could only see the final response after selecting from the candidate set. Table 3 shows pre-selection candidate sets for both beam search and iterative beam search.

Finally, we observe a significant gap between the best search strategy and humans in these diversity metrics. Together with the gap we observed in human evaluation scores, we suspect that the lack of diversity in the output responses is a major factor behind the low performance of the tested neural dialogue model in the human evaluation.

7 Conclusion and Discussion

We have performed realistic human evaluation of the neural dialogue model to validate the importance of exploring better search strategies. We observed that careful design of human evaluation is necessary to properly evaluate the ability of the neural dialogue model to conduct a conversation with a human. The proposed Bayesian calibration of model scores helps to account the annotator bias observed in human evaluation.

Extensive analysis reveals that greedy search, which has been the inference algorithm of choice in neural dialogue modeling, significantly lags behind more sophisticated search strategies such as beam search and iterative beam search.

We have proposed the iterative beam search which produces more diverse set of candidate responses w.r.t. pre-selection distinct-n metric. Post-selection final responses with iterative beam search have higher log-probability compared to other search strategies. In spite of this, there is only a marginal difference between iterative beam search and beam search w.r.t. scores from human evaluation and post-selection distinct-n. This suggests that final response selection strategy is as important as the search strategy being used and can be a major factor in the inference pipeline of the neural dialogue model. We leave improving this strategy to future work.

Finally, the model assigns lower probability to reference responses, which implies suboptimality in the current neural dialogue model . It is necessary in the future to test the proposed search strategies with new models.

Acknowledgments

KC is partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from Pattern Recognition to AI) and Samsung Electronics (Improving Deep Learning using Latent Structure).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. 2012. Diverse m-best solutions in markov random fields. In *European Conference on Computer Vision*, pages 1–16. Springer.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. 2018. Pyro: Deep Universal Probabilistic Programming. *arXiv preprint arXiv:1810.09538*.

Yun Chen, Victor OK Li, Kyunghyun Cho, and Samuel R Bowman. 2018. A stable and effective learning strategy for trainable greedy decoding. *arXiv preprint arXiv:1804.07915*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Dev K Dalal, Nathan T Carter, and Christopher J Lake. 2014. Middle response scale options are inappropriate for ideal point scales. *Journal of Business and Psychology*, 29(3):463–478.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2019. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Xiaodong Gu, Kyunghyun Cho, Jungwoo Ha, and Sunghun Kim. 2018. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew D Hoffman and Andrew Gelman. 2014. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017a. Learning to decode for future success. *arXiv preprint arXiv:1701.06549*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017b. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. *arXiv preprint arXiv:1809.01984*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Alexander H Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018. Improving variational encoder-decoders in dialogue generation. *arXiv preprint arXiv:1802.02032*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David J Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI*.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie SUN, Xiaolong Wang, Zhuoran Wang, and Chao Qi. 2017. Neural response generation via gan with an approximate embedding layer. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 617–626. Association for Computational Linguistics.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. *arXiv preprint arXiv:1804.07754*.

Yury Zemlyanskiy and Fei Sha. 2018. Aiming to know you better perhaps makes me a more engaging dialogue partner. *arXiv preprint arXiv:1808.07104*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.

A Selected dialogue transcripts from the evaluation

We publish dialogue transcripts from human evaluation for reader's analysis. We randomly select a transcript per each search strategy. Formatted representations are printed in tables from Table 4 to Table 6. The second row refers to personalized context from PersonaChat dataset (Zhang et al., 2018). The third row prints the whole dialogue where each turn is bounded with a box. Left column named `Annotator` contains responses written by the annotator. Middle column prints positional ordering of turns. Right column named `Model` contains responses generated by the model. The caption contains the search type and the score given by the annotator. We have prepared all evaluation scripts for reader in additional materials and we encourage everyone to read it.

Annotator		Model
i lay awake at night so i can watch the stars. i m often late for class. i wish i was taller so i could reach the cookie jar. i want a porsche. i cuddle up with my dog every night.	Persona	i like to hike and spend time outdoors. i am close to my mother , but not to my father. my favorite thing to eat is a burrito.
Hello how are you?	0	I ’ m doing well, how are you?
I’m great! Waiting for dark so I can look at the stars!	1	that is cool, I love the outdoors, I love to hike.
My dog and I love hiking, and cuddling!	2	I love hiking, I love hiking, I love hiking.
that’s awesome! I drive to the traiils in my porsche.	3	I have a mustang, I ’ ve a mustang, it is a nice car.
That’s really cool! I love sports cars.	4	I like to go hiking, I love to eat burritos.

Table 4: Search strategy: **greedy**. Annotator score: **3**

Annotator		Model
my father died when i was 2. i have blue eyes. i love fishing. my favorite season is spring.	Persona	i believe in aliens. to free my mind from feeling depressed i look up at the stars at night. i cannot choose between lollipops and rainbows. i would love to be a professional eater and not get fat. i hate cleaning up after my husband and children.
Hey there, how are you?	0	I ’ m good. how are you?
pretty good. I’m planning my next fishing trip!	1	that sounds fun. I ’ m going to be a professional eater
that sounds amazing! I would love that.	2	do you have any hobbies?
I really like going fishing, especially in the spring time.	3	I like to stay at night with my husband and kids
That is nice! I wish I had a family, my dad died when I was really little.	4	I ’ m sorry to hear that

Table 5: Search strategy: **beam**. Annotator score: **4**

Annotator		Model
i eat tuna fish salad at least every day. my favorite music is two steps from hell and rock opera genre. i have sandy brown hair and green eyes. i can t get enough gummy worms to satisfy my sweet tooth. i read sci fi space adventures with a passion.	Persona	i am a yoga instructor. i married my highschool sweetheart. pink is my favorite color. i enjoy being alone.
hello there friend!	0	hello, how are you?
I’m great. how are you?	1	I ’ m doing well, thank you.
What’s your favorite food?	2	I love pizza, what about you?
tuna fish salad is my favorite. i eat it every day.	3	do you have a favorite color? mine is pink.
I like all of the colors, but my eyes are green. do you have any hobbies?	4	I like to go alone.
ah, I enjoy reading sci-fi. that’s something you do alone.	5	do you have any hobbies?
reading. I also really like rock operas.	6	what is your favorite genre?
rock opera. but my favorite band is two steps from hell. what about you?	7	I like all kinds of music.
is there any food you can’t get enough of?	8	I do not have a lot of free time.
ah, sorry to hear that. It was nice chatting with you, though!	9	it was nice talking to you too!

Table 6: Search strategy: **iter-beam**. Annotator score: **3**